**（状态示意图）**

**消息传递**：每轮计算结束后，每个worker都会收集属于自己的fragment的更新结果，这些结果来自与每个fragment内部更新数据的改变量，并将更新结果发送给其他的worker，为实现该消息传递机制，每个worker需声明维护以下数据结构：

1. 图中全部点与该点所属worker的映射，以便顺利将该点的更新消息放松给对应worker
2. 一个缓冲buffer，用于接收来自其他worker发送的消息

MPAP的消息传递过与经典AAP模型相同，是点对点通信、且发生在图计算的任意时刻，即任意worker pi在任意适合都可将消息发送给worker pj而不需要关系pj此时处于状态，同样，worker pi也会在任意时刻接受来自其他worker的消息，保存在缓冲buffer中，该过程不会阻塞任何计算过程。但与AAP模型不同的是，每个fragment除发送本轮更新的消息外，还需像coordicator节点发送本轮的运行时信息，该信息用于后续的模型训练。

**MPAP模型**：

MPAP模型中，为解决同步模型下的“木桶效应”及异步模型下的冗余计算问题，每个计算节点引入了限定值，之后每轮迭代计算开始前，各计算节点需判断是否等待长的时间以积累更多的消息，我们给出基于下述函数进行动态调整：

上述函数中的各参数描述如下，表示某一计算节点：

(1) ：表示计算节点当前接收的消息量。直观上来看，如果 越大，则当前计算节点接收消息越多，应尽快开始下轮迭代计算。

(2) ：表示本轮迭代计算开始前期望接收的消息量。该变量定义如下：，其中可由用户自定义初始化，为了尽快减少冗余计算；为消息到达速率；是 的一部分，后者为运行时间。

(3) ：表示计算节点为接收到期望的消息量而等待的时间，定义为；表示计算节点前一轮计算后的空闲时间，以防止无限期的等待。

**终止条件**：MPAP模型的终止条件与经典AAP模型相同，即新一轮迭代计算开始前，如果当前没有任何接收到的消息，则会像coordinator发送inactive flag, 当coordinator接收到所有worker的inactive flag后，会广播各个worker 终止命令，各个worker会对此返回ack确认自己是否inactive，如果某个worker仍有任务需要计算，则回应wait命令，继续进行下一次计算，coordicator重新接入监听等待状态。